Indian Institute of Technology Kanpur
Estimation Methods in
Python / MATLAB
Extended Kalman Filter (EKF)

Mahesh Pandurang Mohite
Sustainable Energy Engineering
22129009

Task: 1

Implementation of EKF algorithm on Differential drive robot [Python]

Solution:

1. The state equation is

$$\begin{bmatrix} x_t \\ y_t \\ \gamma_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \gamma_{t-1} \end{bmatrix} + \begin{bmatrix} cos(\gamma_{t-1}) * dt & 0 \\ sin(\gamma_{t-1}) * dt & 0 \\ 0 & dt \end{bmatrix} \begin{bmatrix} v_{t-1} \\ \omega_{t-1} \end{bmatrix} + \mathbf{w}_{t-1}$$

$$\begin{bmatrix} x_t \\ y_t \\ \gamma_t \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} x_{t-1} + v_{t-1} * cos(\gamma_{t-1}) * dt \\ y_{t-1} + v_{t-1} * sin(\gamma_{t-1}) * dt \\ \gamma_{t-1} + \omega_{t-1} * dt \end{bmatrix}$$

2. Output equation is

$$y_t = Hx_t + w_t$$

3. Initial values given are

$$\begin{bmatrix} x^1(0) \\ x^2(0) \\ x^3(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \qquad P(0) = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}, \qquad u(0) = \begin{bmatrix} 4.5 \\ 0 \end{bmatrix}$$

4. Results for each time step are:

   (a) `Timestep k=1`
   - State Estimate Before EKF=[4.51 0.01 0.003]
   - Observation=[4.721 0.143 0.006]
   - State Estimate After EKF=[4.5839 0.043 -0.0164]

   (b) `Timestep k=2`
   - State Estimate Before EKF=[9.0933 -0.0207 -0.0134]
   - Observation=[9.353 0.284 0.007]
   - State Estimate After EKF=[9.2078 0.121 -0.0252]

   (c) `Timestep k=3`
   - State Estimate Before EKF=[13.7164 0.0175 -0.0222]
   - Observation=[14.773 0.422 0.009]
   - State Estimate After EKF=[14.3241 0.2235 -0.0276]

   (d) `Timestep k=4`
   - State Estimate Before EKF=[18.8324 0.1092 -0.0246]
   - Observation=[18.246 0.555 0.011]
   - State Estimate After EKF=[18.4269 0.3413 -0.0273]

(e) `Timestep k=5`

- State Estimate Before EKF=[22.9352 0.2284 -0.0243]
- Observation=[22.609 0.715 0.012]
- State Estimate After EKF=[22.6904 0.4858 -0.0266]

Task: 2

1. Object tracking in 2-D plane (4 states)
2. Object tracking in 3-D plane (6 states) [MATLAB]

Solution:

2-D

1. The discrete time state equation is given by (Motion model) :

$$\mathbf{x}_k = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \mathbf{w}_{k-1} = \begin{bmatrix} \texttt{position in x-direction} \\ \texttt{velocity in x-direction} \\ \texttt{position in y-direction} \\ \texttt{velocity in y-direction} \end{bmatrix} = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix}$$

2. Observation model is given by (Output equation) :

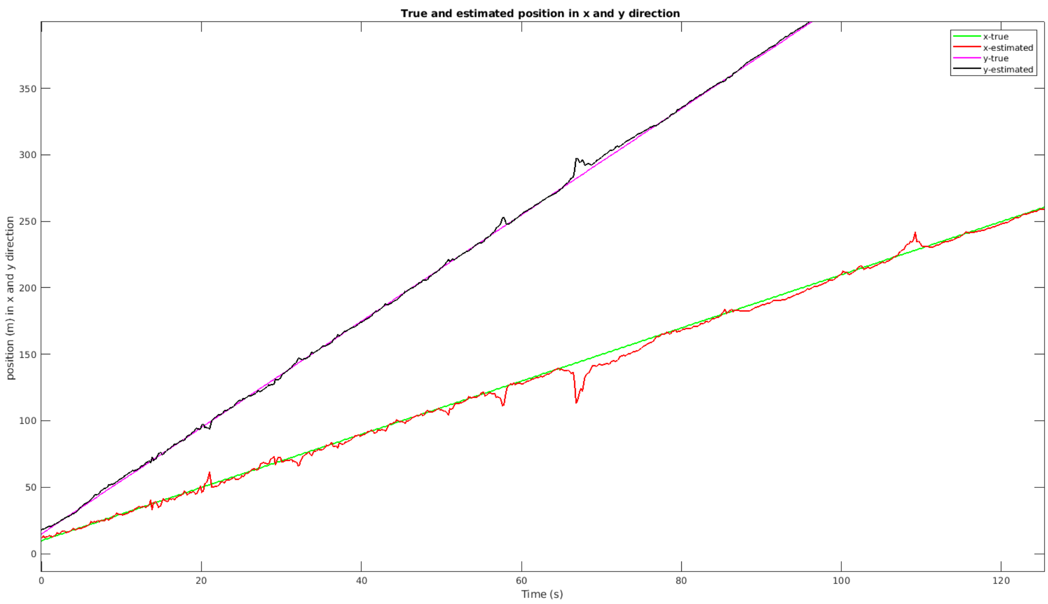
$$z_k = \begin{bmatrix} \sqrt{x_k^2 + y_k^2} \\ \tan^{-1}(\frac{y_k}{x_k}) \end{bmatrix} = \begin{bmatrix} r_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} \texttt{range} \\ \texttt{theta} \end{bmatrix}$$

3. Jacobian calculation of Measurement model, that is H matrix :

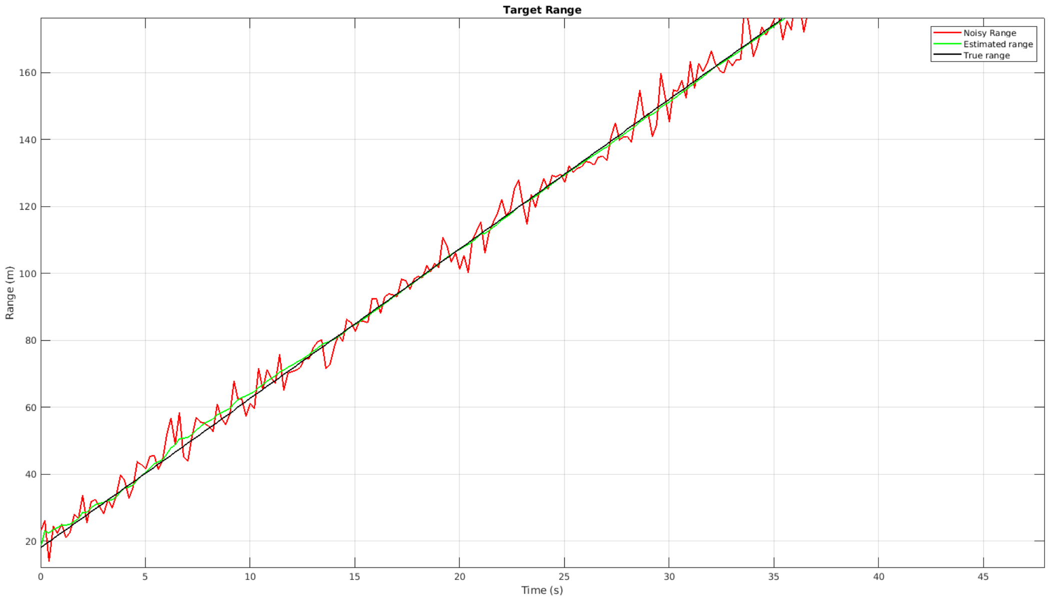
$$H = \frac{\partial z}{\partial x} = \begin{bmatrix} \frac{\partial r}{\partial x} & \frac{\partial r}{\partial \dot{x}} & \frac{\partial r}{\partial y} & \frac{\partial r}{\partial \dot{y}} \\ \frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial \dot{x}} & \frac{\partial \theta}{\partial y} & \frac{\partial \theta}{\partial \dot{y}} \end{bmatrix}$$

$$H = \begin{bmatrix} \frac{x}{\sqrt{x^2+y^2}} & 0 & \frac{y}{\sqrt{x^2+y^2}} & 0 \\ \frac{-y}{x^2+y^2} & 0 & \frac{x}{x^2+y^2} & 0 \end{bmatrix}$$

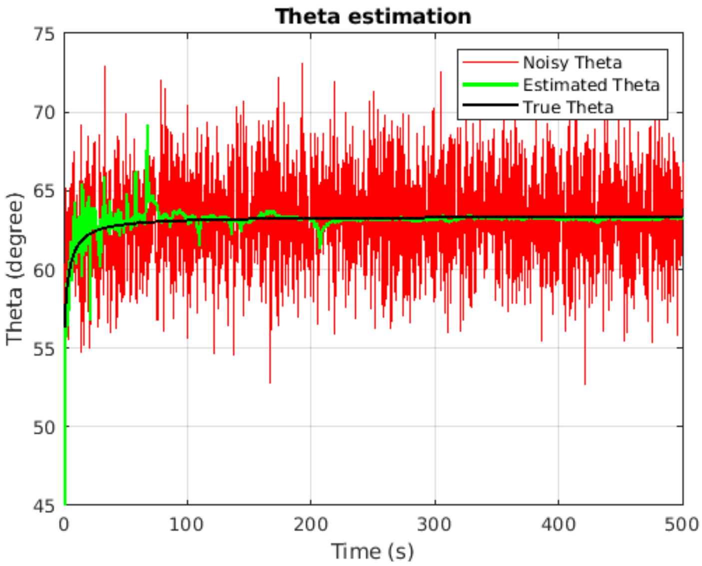
4. Results of EKF-estimation are:
   True and estimated position in x and y direction



Target range



Theta estimation



3-D

(a) The discrete time state equation is given by (Motion model) :

$$\mathbf{x}_k = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \mathbf{w}_{k-1} = \begin{bmatrix} \texttt{position in x-direction} \\ \texttt{position in y-direction} \\ \texttt{position in z-direction} \\ \texttt{velocity in x-direction} \\ \texttt{velocity in y-direction} \\ \texttt{velocity in z-direction} \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}$$

(b) Observation model is given by (Output equation) :

$$z_k = \begin{bmatrix} \sqrt{x_k^2 + y_k^2 + z_k^2} \\ \tan^{-1}(\frac{y_k}{x_k}) \\ tan^{-1}\left(\frac{z}{\sqrt{x_k^2+y_k^2+z_k^2}}\right) \end{bmatrix} = \begin{bmatrix} r_k \\ \theta_k \\ \epsilon_k \end{bmatrix} = \begin{bmatrix} \texttt{range} \\ \texttt{azimuth} \\ \texttt{elevation} \end{bmatrix}$$

(c) Jacobian calculation of Measurement model, that is H matrix :

$$H = \frac{\partial z}{\partial x} = \begin{bmatrix} \frac{\partial r}{\partial x} & \frac{\partial r}{\partial y} & \frac{\partial r}{\partial z} & 0 & 0 & 0 \\ \frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial y} & \frac{\partial \theta}{\partial z} & 0 & 0 & 0 \\ \frac{\partial \epsilon}{\partial x} & \frac{\partial \epsilon}{\partial y} & \frac{\partial \epsilon}{\partial z} & 0 & 0 & 0 \end{bmatrix}$$

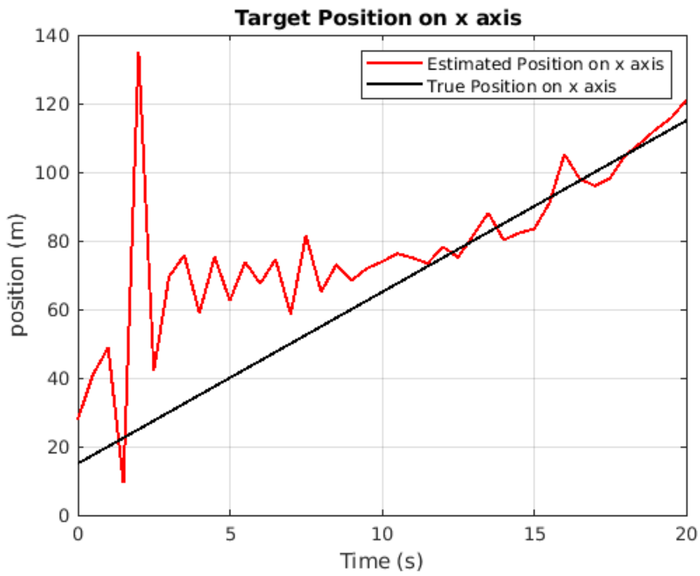
$$H = \frac{\partial z}{\partial x} = \begin{bmatrix} \frac{x}{\sqrt{x^2+y^2+z^2}} & \frac{y}{\sqrt{x^2+y^2+z^2}} & \frac{z}{\sqrt{x^2+y_k^2+z_k^2}} & 0 & 0 & 0 \\ \frac{y}{\sqrt{x^2+y^2}} & \frac{x}{\sqrt{x^2+y^2}} & 0 & 0 & 0 & 0 \\ \frac{-xz}{D^2\sqrt{x^2+y^2}} & \frac{-yz}{D^2\sqrt{x^2+y^2}} & \frac{\sqrt{x^2+y^2}}{D^2} & 0 & 0 & 0 \end{bmatrix}$$

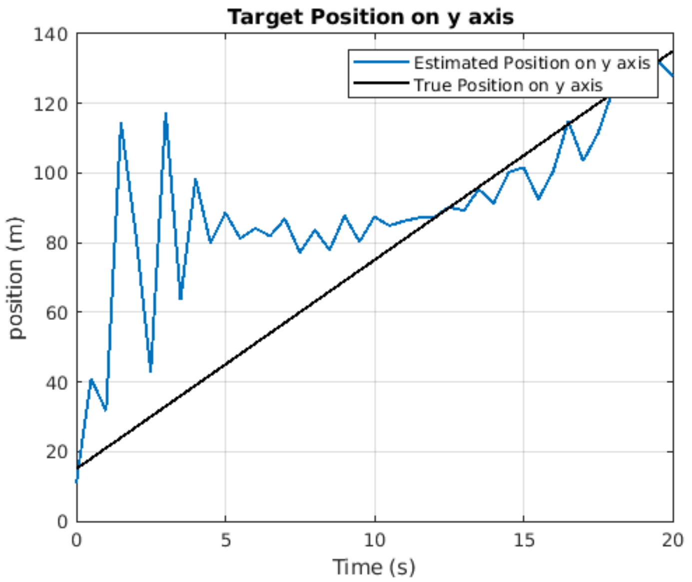
(d) $D = \sqrt{x_k^2 + y_k^2 + z_k^2}$

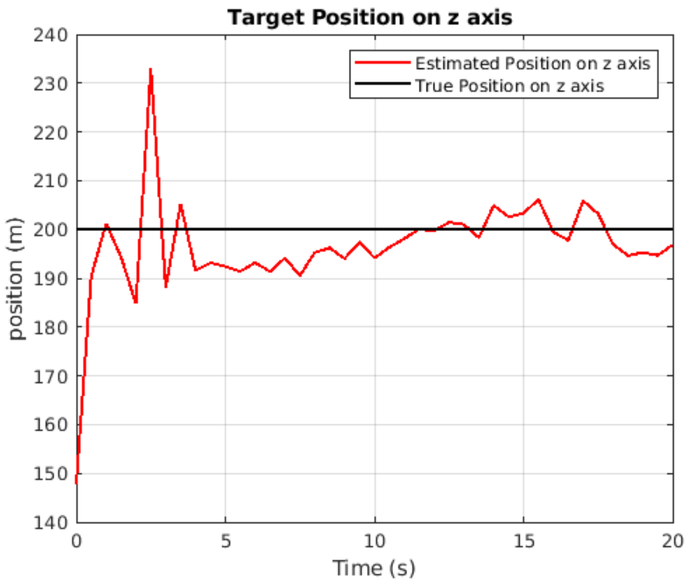
(e) Mean and variance of measured noise is taken as Range(0,8) , Azimuth(0,2) , Elevation(0,0.5) which implies

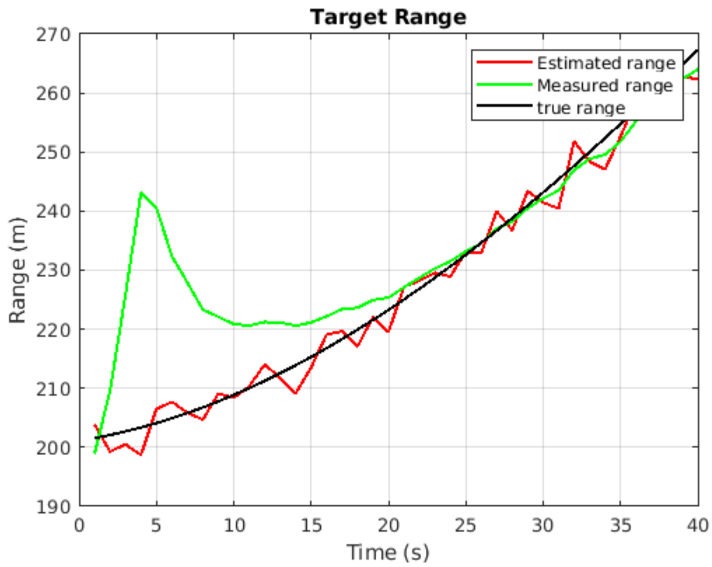
$$R = \begin{bmatrix} 8 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}$$

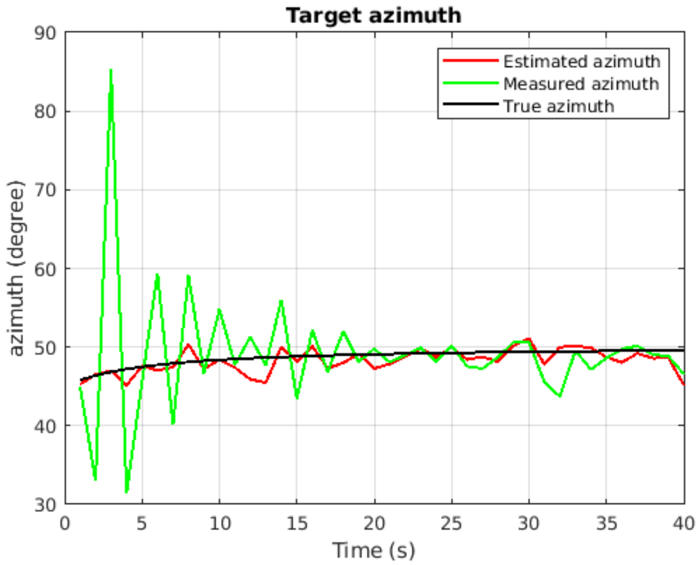
(f) Results of EKF-estimation are:

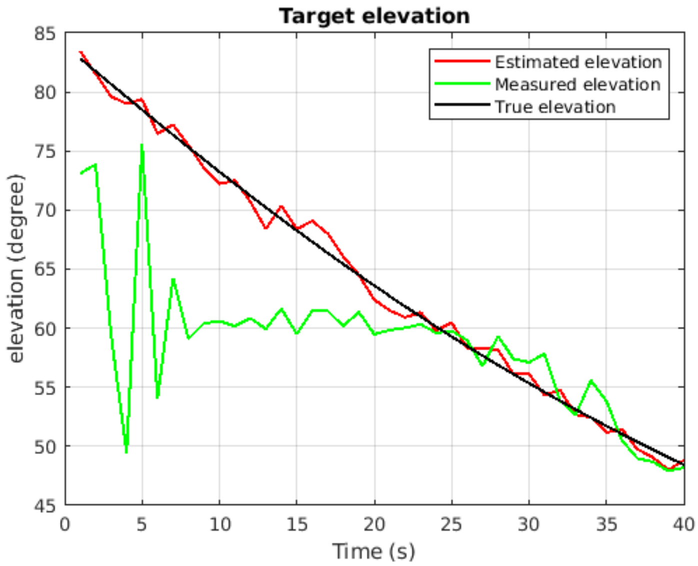



Target Position on y axis
Estimated Position on y axis
True Position on y axis
position (m)
Time (s)
0
20
40
60
80
100
120
140
0
5
10
15
20


Target Position on z axis
Estimated Position on z axis
True Position on z axis
position (m)
Time (s)
140
150
160
170
180
190
200
210
220
230
240
0
5
10
15
20


Target Range
Estimated range
Measured range
true range
Range (m)
Time (s)
190
200
210
220
230
240
250
260
270
0
5
10
15
20
25
30
35
40


Target azimuth
Estimated azimuth
Measured azimuth
True azimuth
azimuth (degree)
Time (s)


Target elevation
Estimated elevation
Measured elevation
True elevation
elevation (degree)
Time (s)

Task: 3

Dynamic model parameter estimation problem , EKF estimation of oscillator position, velocity, and damping factor (linear damped oscillator) [MATLAB]

Solution:

1. Linear dynamic equation in continuous time is nonlinear which is given by:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} x_2 \\ -\omega^2 x_1 - 2x_2x_3 + \omega \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} w(t)$$

2. The observation equation (output equation) is linear :

$$z(t) = x_1(t) + v(t)$$

3. $\omega = 10$, $\zeta = 0.1$ (True value)

4. Initial conditions are taken as

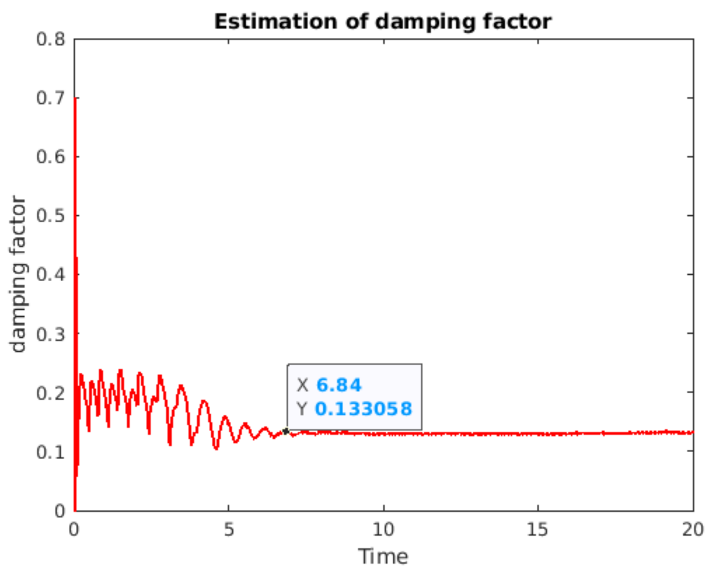
$$\begin{bmatrix} x^1(0) \\ x^2(0) \\ x^3(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0.7 \end{bmatrix}, \qquad P(0) = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

5. Q = 4.47 , R = 0.001

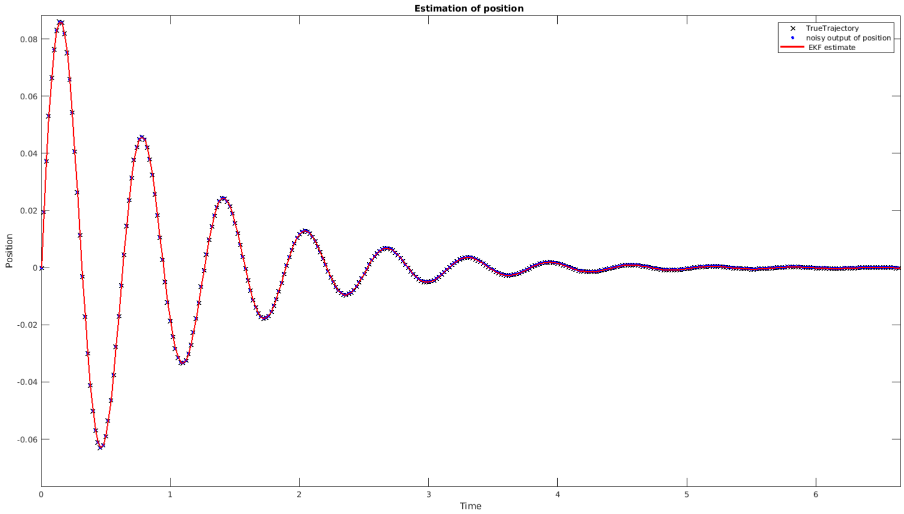
6. Equivalent discrete non-linear plant and linear observation model are

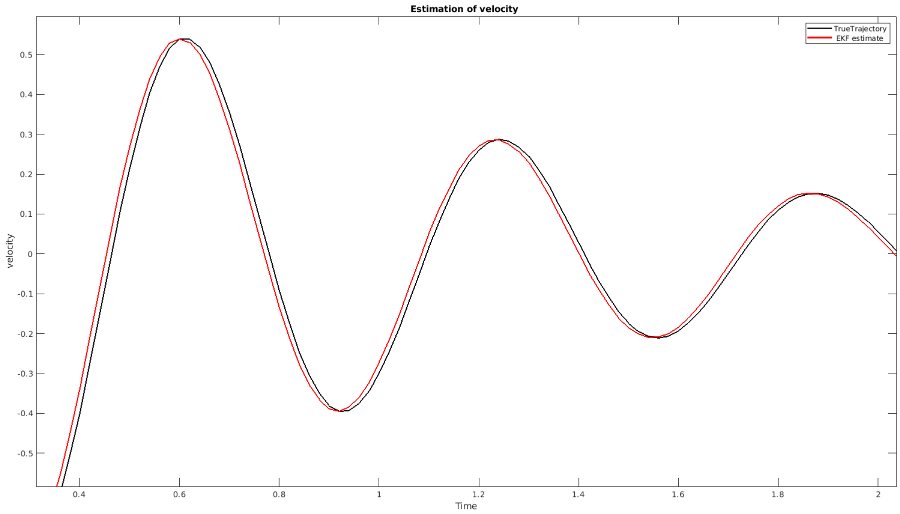
$$\begin{aligned} x_k^1 &= x_{k-1}^1 + T x_{k-1}^2 \\ x_k^2 &= -100T x_{k-1}^1 + (1 - 20T x_{k-1}^3) x_{k-1}^2 + T w_{k-1} \\ x_k^3 &= x_{k-1}^3 \\ z_k &= x_k^1 + v_k \end{aligned}$$

7. Damping factor result, the true value of zeta was 0.1 and the EKF-estimate of zeta was 0.133.



8. The EKF-estimates of position and velocity were within error 10^{-3} and 0.07 respectively. Below are the curves





Task: 4

1. EKF application to system with 2 states with motion model f(x) (non-linear)
2. EKF application to system with both motion(state equation) and observation model(output equation) non-linear with 3 states (2 examples) [MATLAB]

Solution:

- Problem-1

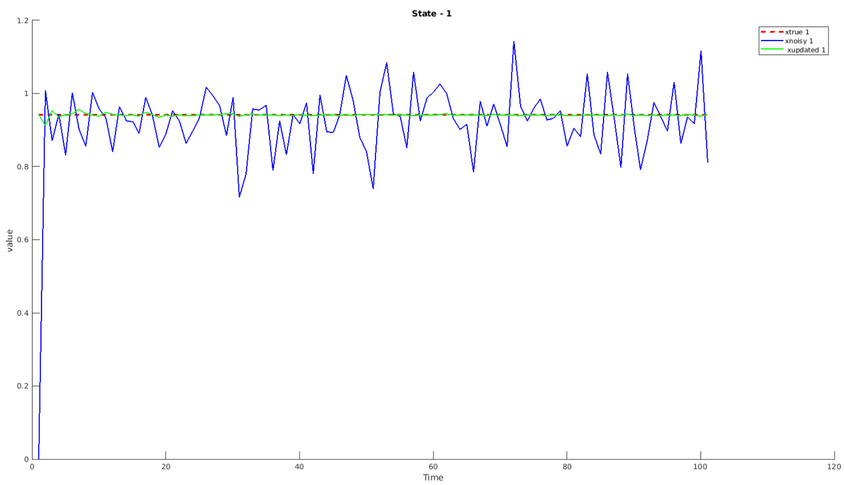
1. State equation is

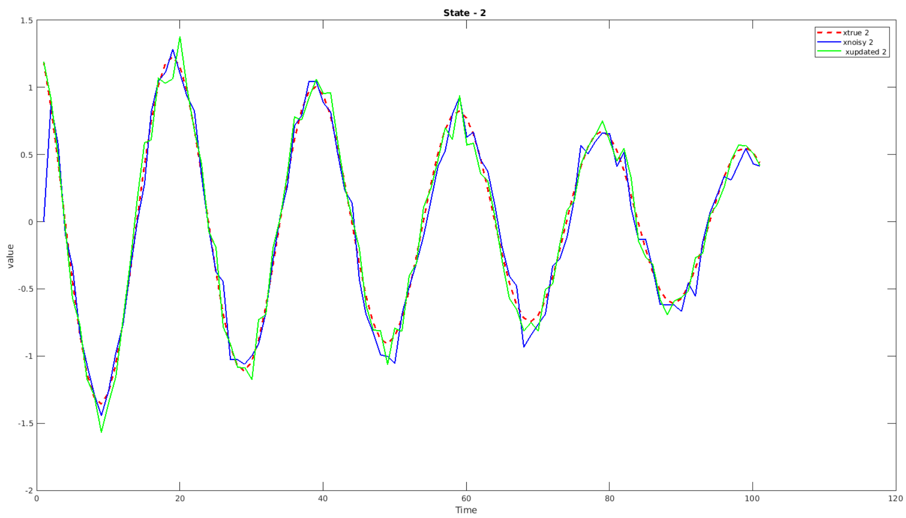
$$x_k = [x^1_{k-1} \ , x^1_{k-1} * x^2_{k-1} \ , x^1_{k-1} * x^3_{k-1}]' = f(x_{k-1})$$

2. Output equation is (State 2 is measured)

$$y_k = x^2_{k-1} * x^2_{k-1}$$

3. Q = 0.1 r = 0.1
4. Results are :





- Problem-2

1. State equation is

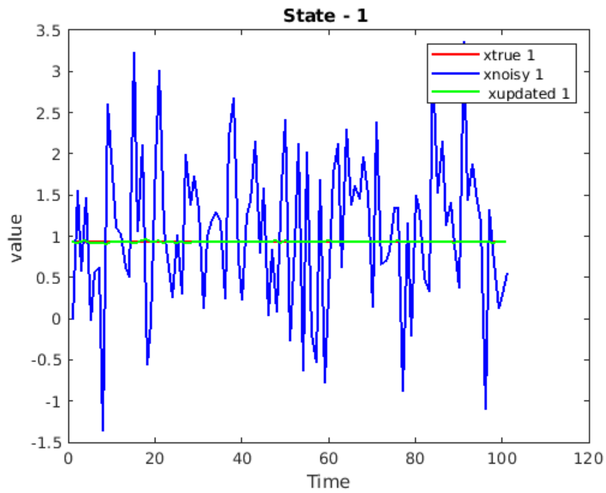
$$x_k = \left[x^1_{k-1}\ ,x^1_{k-1} * x^2_{k-1}\right]' = f(x_{k-1})$$

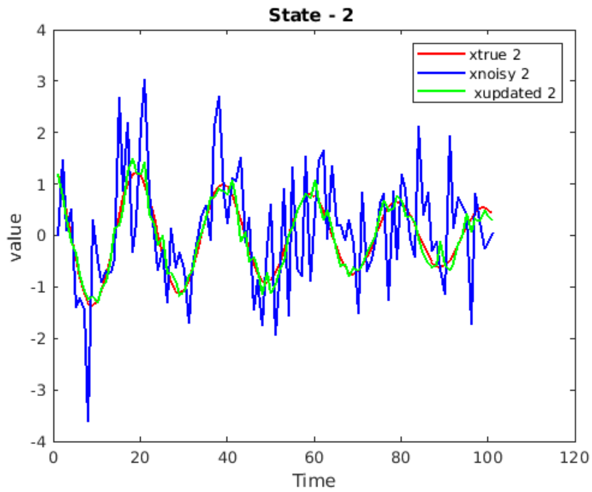
2. Output equation is (State 2 is measured)

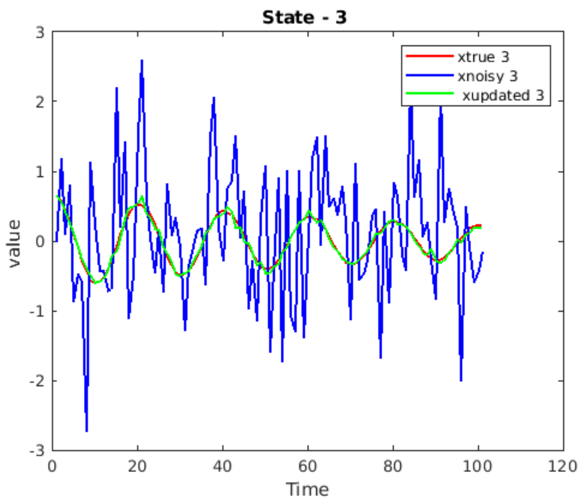
$$y_k = x^2_{k-1}$$

3. Q = 0.1 r = 0.1

4. Results are :







- Problem-3

1. State equation is

$$x_k = [x^1_{k-1}\ , x^1_{k-1} * x^2_{k-1}\ , x^3_{k-1} * x^3_{k-1}]' = f(x_{k-1})$$

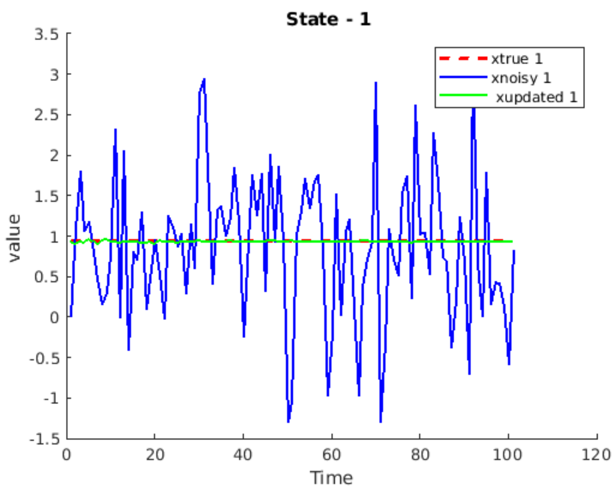
2. Output equation is (State 2 is measured)

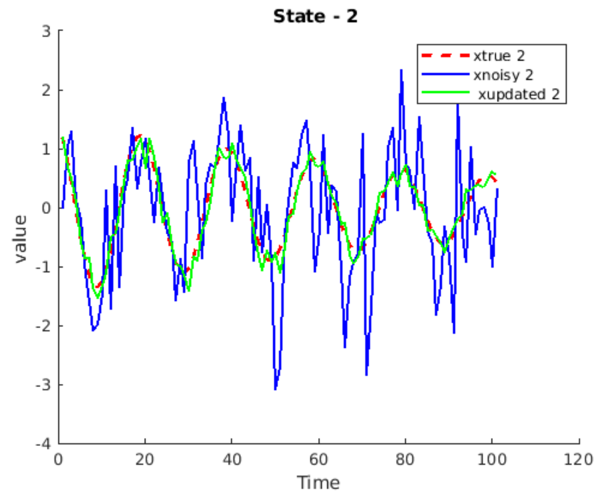
$$y_k = x^2_{k-1} * x^2_{k-1}$$

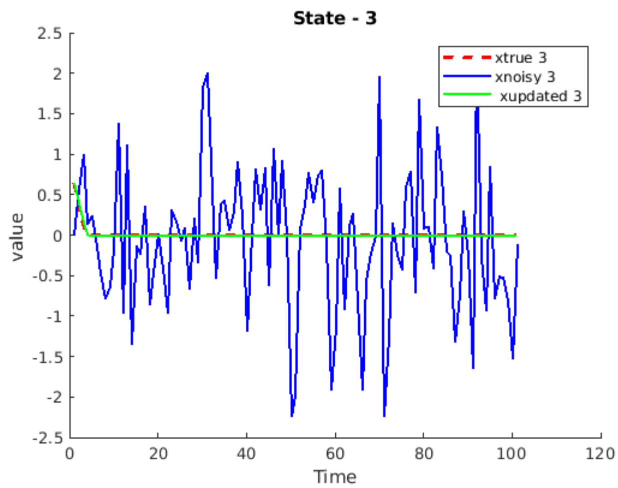
3. Q = 0.9   r = 0.6
4. Matrix H is given by

$$H = \begin{bmatrix} 0 & 2x^2 & 0 \end{bmatrix}$$

5. Results are :




State - 2
xtrue 2
xnoisy 2
xupdated 2
value
Time
3
2
1
0
-1
-2
-3
-4
0
20
40
60
80
100
120


State - 3
xtrue 3
xnoisy 3
xupdated 3
value
Time
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
0
20
40
60
80
100
120

Task: 5

Estimation of parameters of a damped mass spring system using extended kalman filter with only one noisy position sensor. [2 states and 2 parameters are estimated]

Solution:

1. Linear dynamic equation in continuous time is nonlinear which is given by $\dot{x} = f(x_1, x_2, x_3, x_4)$:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \end{bmatrix} = \begin{bmatrix} x_2 \\ -x_4x_2 - x_3x_1 \\ 0 \\ 0 \end{bmatrix}$$

2. x_1 and x_2 are the states and x_3 and x_4 are parameters. Description is mentioned below

$$x_1 = \text{Position}$$
$$x_2 = \text{Velocity}$$
$$x_3 = \text{Spring constant}$$
$$x_4 = \text{Damping coefficient}$$

3. The observation equation (output equation) is linear :

$$z(t) = x_1(t) + v(t)$$

4. Only the position is being measured, Hence,

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

5. Initial conditions are taken as

$$\begin{bmatrix} x^1(0) \\ x^2(0) \\ x^3(0) \\ x^4(0) \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 6 \\ 8 \end{bmatrix}, \qquad P(0) = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}$$

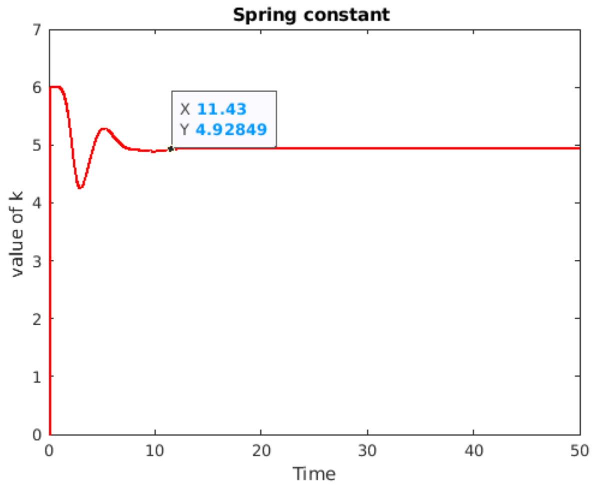
6. Q = 0.01 , R = 0.1

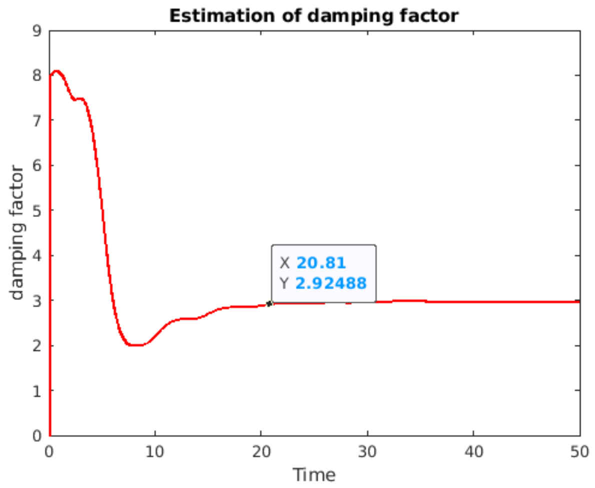
7. Non-linear observability is checked with the help of high ordered Lie derivatives with respect to state. The dynamics satisfies the non-linear observability test.

8. Equivalent discrete non-linear plant is

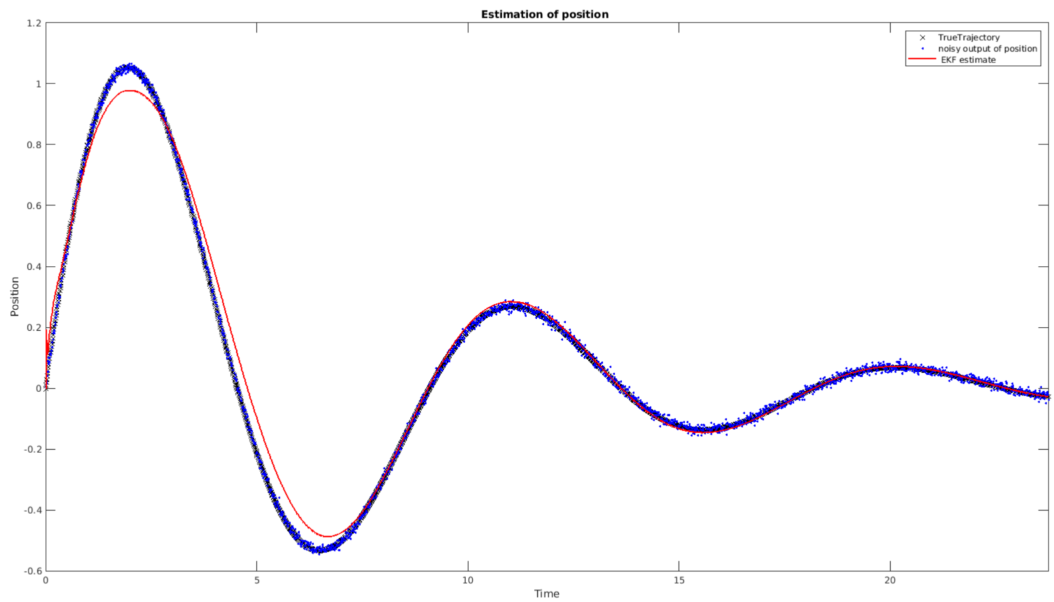
$$x_k^1 = x_{k-1}^1 + Tx_{k-1}^2$$
$$x_k^2 = x_{k-1}^2 + T(-x_{k-1}^4 x_{k-1}^1 - x_{k-1}^3 x_{k-1}^2)$$
$$x_k^3 = x_{k-1}^3$$
$$x_k^4 = x_{k-1}^4$$

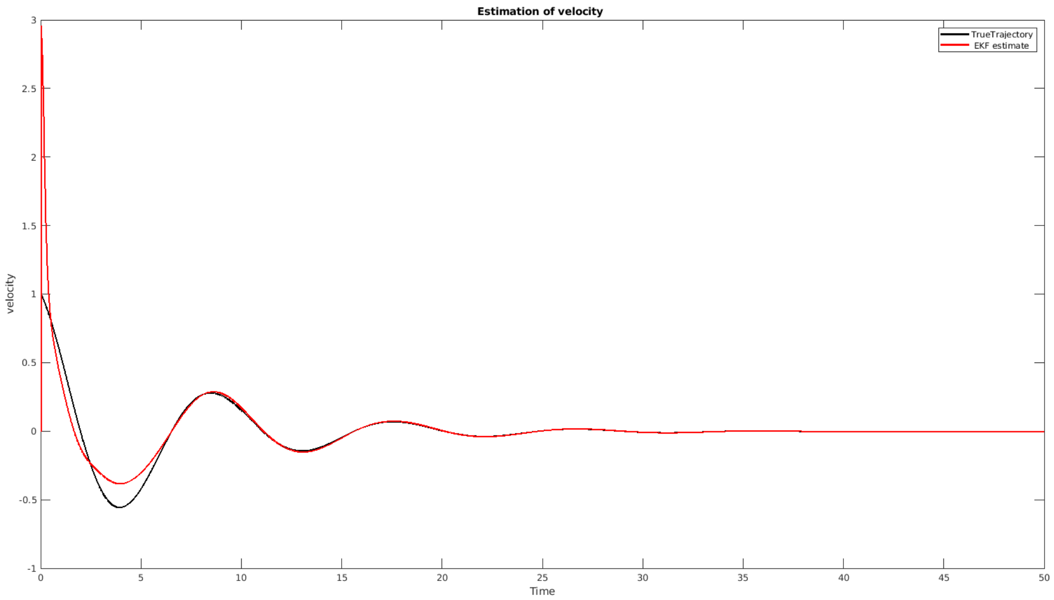
9. True values of spring constant $k = 5$, and damping factor $b = 3$. EKF estimates results are :





10. EKF estimates of Position and velocity are :





Estimation of velocity
TrueTrajectory
EKF estimate
velocity
Time
3
2.5
2
1.5
1
0.5
0
-0.5
-1
0
5
10
15
20
25
30
35
40
45
50

Task: 6

1. Tracking an airplane (Obtaining the EKF-estimates of position , velocity and altitude)[Python]

Solution:

1. The state variable matrix is

$$\mathbf{x} = \begin{bmatrix} x \\ \dot{x} \\ y \end{bmatrix} = \begin{bmatrix} \texttt{distance} \\ \texttt{velocity} \\ \texttt{altitude} \end{bmatrix}$$

2. The State Transition Matrix (STM) is

$$\mathbf{F} = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3. Measurement function is (measurement of slant range distance) :

$$h(\mathbf{x}) = \sqrt{x^2 + y^2}$$

4. Calculating partial derivative of h w.r.t. state variables, we get $\mathbf{H}$ matrix as

$$\mathbf{H} = \begin{bmatrix} \frac{\partial h}{\partial x} & \frac{\partial h}{\partial \dot{x}} & \frac{\partial h}{\partial y} \end{bmatrix}$$

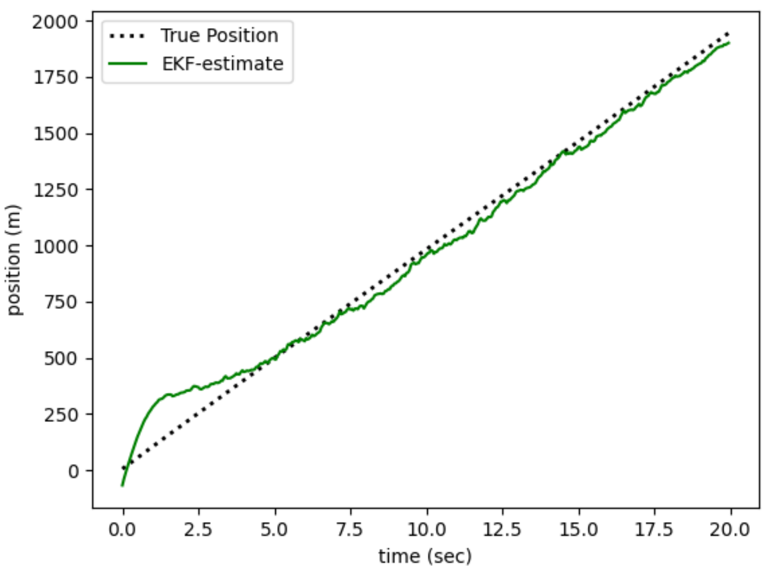
$$\mathbf{H} = \begin{bmatrix} \frac{x}{\sqrt{x^2+y^2}} & 0 & \frac{y}{\sqrt{x^2+y^2}} \end{bmatrix}$$

5. R = 25 , $\sigma_{\text{range}} = 5$, Q = 0.1*I

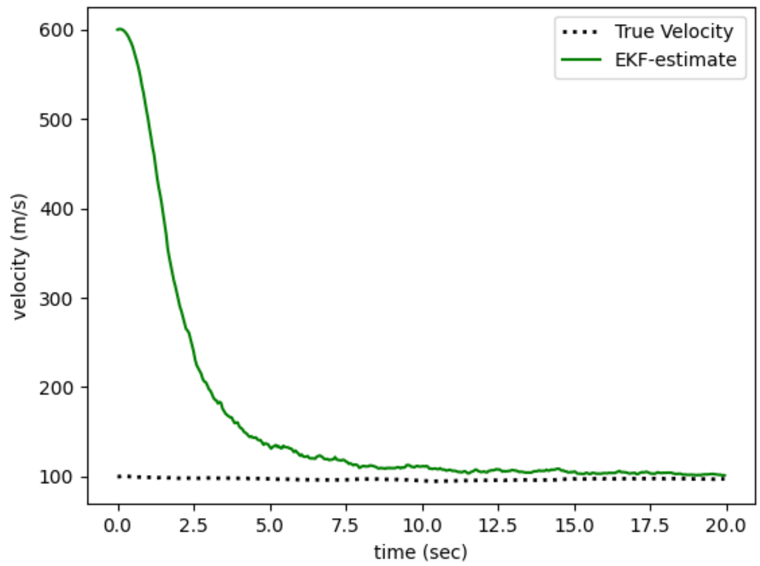
6. Initial values given are :

$$\begin{bmatrix} x^1(0) \\ x^2(0) \\ x^3(0) \end{bmatrix} = \begin{bmatrix} -100 \\ 600 \\ 2000 \end{bmatrix}, \qquad P(0) = \begin{bmatrix} 50 & 0 & 0 \\ 0 & 50 & 0 \\ 0 & 0 & 50 \end{bmatrix}$$

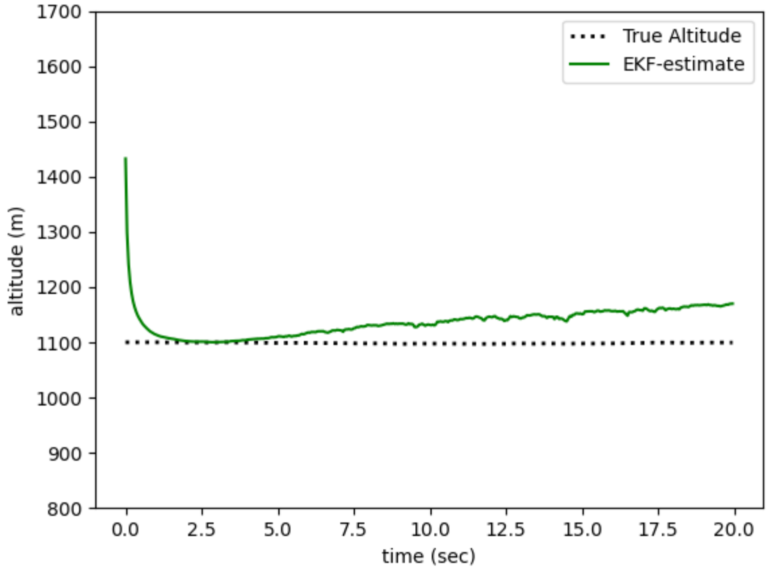
7. The results of the implementation are:
   Position:



Velocity:



Altitude:



Task: 7

1. Recursively estimate the position of a vehicle along a trajectory [Python]

Solution:
Description of the problem given :

1. The vehicle is equipped with a very simple type of LIDAR sensor, which returns range and bearing measurements corresponding to individual landmarks in the environment.

2. The global positions of the landmarks are assumed to be known beforehand. [Assumption]

3. We also know which measurement belong to which landmark.[Known data association](Assumption)

4. Motion model :

$$\mathbf{x}_k = \mathbf{x}_{k-1} + T \begin{bmatrix} cos(\theta_{k-1}) & 0 \\ sin(\theta_{k-1}) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_k \\ \omega_k \end{bmatrix} + \mathbf{w}_k$$

5. Description of parameters (Motion model) :

   (a) $\mathbf{x}_k = [x\ y\ \theta]^T$ is the current 2D pose of the vehicle
   (b) $v_k$andω_k are the linear and angular velocity odometry readings, which we use as inputs to the model
   (c) The process noise $\mathbf{w}_k$ has a (zero mean) normal distribution with a constant covariance Q .

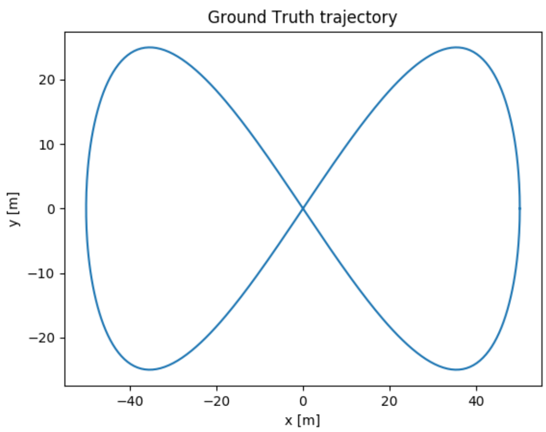
6. Measurement model :

$$\mathbf{y}_k = \begin{bmatrix} \sqrt{(x_l - x_k - dcos(\theta_k))^2 + (y_l - y_k - dsin(\theta_k))^2} \\ atan2(y_l - y_k - dsin(\theta_k), x_l - x_k - dcos(\theta_k) - \theta_k) \end{bmatrix} + \mathbf{v}_k$$

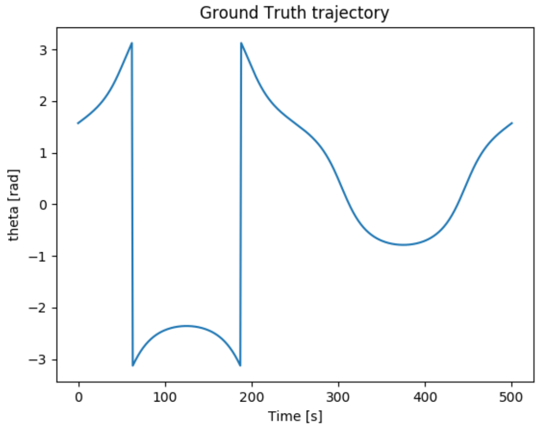
7. Description of parameters (measurement model) :

   (a) $x_l$andy_l are the ground truth coordinates of the landmark l
   (b) x_k, y_k, θ_k represent the current pose of the vehicle
   (c) d is the known distance between robot center and laser rangefinder (LIDAR)
   (d) The landmark measurement noise $\mathbf{v}_k$ has a (zero mean) normal distribution with a constant covariance R.

8. The Ground truth position and orientation is provided
   Position:

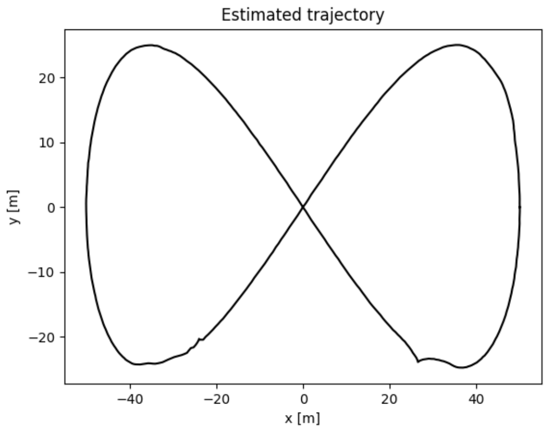


Theta :



Implementing EKF Algorithm

1. Initial values are taken as
$$\begin{bmatrix} x^1(0) \\ x^2(0) \\ x^3(0) \end{bmatrix} = \begin{bmatrix} 50 \\ 0 \\ 1.5707 \end{bmatrix}, \qquad P(0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$$
2. Q = $0.01\mathbf{I}$, R = $0.01\mathbf{I}$
3. EKF - estimates of position and orientation are :
Position:



Theta :